Amendments to the Claims

1. (Amended): A method of converting a non-object oriented computer environment to a new object oriented computer environment, the method comprising the steps of:

identifying ~~an~~ the existing object oriented computer environment;

identifying the non-object oriented computer environment;

defining requirements for the new object oriented computer environment;

selecting grammar and syntax compatible with the non-object oriented computer environment;

developing object oriented extensions, wherein an existing application of the non-object oriented computer environment remains executable and wherein the new object oriented computer environment accesses information of the non-object oriented computer environment; and,

preparing the new object oriented computer environment, wherein the new object oriented computer environment includes ~~the~~, requirements ~~the~~, grammar ~~and~~, syntax and object oriented extensions.

2. (Original): The method of Claim 1, wherein the step of identifying an existing object oriented computer environment includes identifying a commercially available object oriented computer environment.

3. (Original): The method of Claim 1, wherein the step of identifying the non-object oriented computer environment includes identifying a legacy non-object oriented computer environment.

4. (Original): The method of Claim 3, wherein the legacy non-object oriented computer environment includes a user language interface and data structures.

5. (Original): The method of Claim 3, wherein the legacy non-object oriented computer environment allows multiple users.

6. (Original): The method of Claim 3, wherein the legacy non-object oriented computer environment includes a distributed environment.

7. (Original): The method of Claim 1, wherein the non-object oriented computer environment allows simulation modeling.

8. (Original): The method of Claim 6, wherein the non-object oriented computer environment allows simulation modeling for the analysis of the performance software executing in a computer system.

9. (Original): The method of Claim 1, wherein the step of selecting grammar and syntax includes selecting the semantics of the non-object oriented computer environment.

10. (Original): The method of Claim 1, wherein the step of selecting grammar and syntax includes selecting semantics compatible to the non-object oriented computer environment.

11. (Original): The method of Claim 1, wherein the step of selecting grammar and syntax includes selecting the semantics of the existing object oriented computer environment.

12. (Original): The method of Claim 1, wherein the step of developing object oriented extensions includes developing an object header structure and an object data structure.

13. (Amended): The method of Claim ~~11~~ 12, wherein the step of developing an object header structure includes developing an object header structure that provides a unified object oriented interface to a user and internal objects.

14. (Amended): The method of Claim ~~11~~ 12, wherein the step of developing an object data structure includes developing an object data structure containing a data structure of the non-object oriented computer environment.

15. (Original): The method of Claim 1 further comprising the step of developing general-purpose utility classes.

16. (Original): The method of Claim 1, wherein the step of preparing the new object oriented computer environment includes creating new code.

17. (Original): The method of Claim 1, wherein the step of preparing the new object oriented computer environment includes creating an operating system.

18. (Original): The method of Claim 1, wherein the new object oriented computer environment includes an object oriented computer language.

19. (Original): A computer system for simulation modeling, the computer system comprising:

an object oriented programming language;

application software written in the object oriented programming language, wherein the application software simulates computer systems;

an application logic function;

data types and scope, wherein the data types and scope include data types and scope of a non-object oriented programming language; and,

a class for message instancing.

20. (Original): The computer system of Claim 19 wherein the object oriented programming language further comprises:

client workload models;

server process infrastructure;

operating system models;

statistics capability;

utility classes; and,

garbage collection.

21. (Original): The computer system of Claim 20 wherein the object oriented programming language is Object Oriented ADN.